

Informe de re-entrega

Equipo: Las Tres Vinchuecas

1. Opinión y validación de muestras.

Con respecto al agregado de opiniones, se redefinió el mensaje `#agregarOpinion` de la clase `Muestra`. En consecuencia, cuando una muestra está votada por experto, ahora se delega en el usuario la responsabilidad de decidir qué acción debe realizar la muestra dependiendo de sus estados.

En cuanto a las validaciones, ahora se delegan en el estado actual de la muestra. El mensaje `#validarMuestra` es enviado por un usuario con estado experto cuando opina sobre una muestra.

2. Estados de las muestras

a. ResultadoActual

Este punto de la corrección ya estaba contemplado en la resolución del trabajo práctico. La clase `Muestra` cuenta con el mensaje `#verificarMuestra()` el cual delega tal responsabilidad a su estado actual. Cuando la muestra se encuentra en estado `"EstadoDeMuestraVotada"`, se encarga del pasaje hacia `"EstadoDeMuestraVotadaPorExperto"` y borra todas las opiniones anteriores de la muestra.

En consecuencia, las únicas opiniones que "quedarán", serán las de los usuarios con estado experto.

4. Organizaciones y zonas.

Se modificó la manera de notificar a la organizaciones ampliando el protocolo de las clases `Organización` y `ZonaDeCobertura`.

`Organización` cuenta ahora con dos mensajes para recibir la información necesaria dependiendo de si se validó o registró una muestra. Lo mismo

sucede en la ZonaDeCobertura pero para informar. El tipo de observer que se implementó utiliza push model.

5. Filtros para las búsquedas.

Se agregó que la clase FiltroPorFecha implemente la interfaz IFiltro y se corrigió el uso de super en métodos heredados.

6. Manejo de recategorizaciones de usuarios (expertos, expertos ad-hoc, comunes)

Para que un usuario novato pase a ser especialista, se decidió que la clase usuario especialista (antes subclase de Usuario), pase a ser un estado del usuario, subclase de EstadoDeUsuarioExperto y redefina el mensaje #actualizarCategoria(). Por lo tanto, la clase Usuario dejó de ser una clase abstracta y ahora puede comportarse de manera polimórfica con respecto a #actualizarCategoria().

Agregamos en la clase AplicacionWeb el mensaje #actualizarCategoriaDeUsuariosALaFecha() el cual permite corroborar que los usuarios se comportan de manera polimórfica con respecto al mensaje #actualizarCategoria().